AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1-35 (cancelled)

36. (Currently Amended) In a computing system, a method of tracking differences between software builds of a software product, the method comprising:

scanning a first binary software build of a software product to automatically discover classes in the first binary software build, wherein the first binary build comprises a first set of binary files compiled from source code of the software product;

discovering dependencies of the discovered classes;

storing a first dictionary data structure including the discovered classes with their corresponding discovered dependencies;

comparing the first dictionary data structure with a second dictionary data structure, the second dictionary data structure including classes organized with their dependencies of a second binary software build, wherein the second binary build comprises a second set of binary files compiled from source code of the software product by comparing classes and dependencies set forth in the first and second dictionary data structures; and

based on the comparison of the first dictionary data structure with the second dictionary data structure, providing an indication of differences between the first and second binary software builds including:

determining what code produced the differences between the first and second binary software builds; and

then generating ~~a listing of~~ code, from intermediate code, that ~~produced~~ specified the differences between the first and second binary software builds.

37. (Previously Presented) The method of claim 36, wherein discovering classes comprises:

(a) loading configuration information for the first binary software build;

(b) identifying a class of the first software build;

(c) creating an instance of the identified class; and

(d) repeating acts (b) and (c) for other classes in the first binary software build.

38. (Previously Presented) The method of claim 37, wherein loading configuration information for the first binary software build comprises loading default configuration information.

39. (Previously Presented) The method of claim 37, wherein loading configuration information for the first binary software build comprises loading additional configuration information.

40. (Previously Presented) The method of claim 37, wherein creating an instance of the identified class comprises checking to see if a factory is defined for the class.

41. (Previously Presented) The method of claim 36, wherein discovering dependencies of the discovered classes comprises exploring the class to determine if there are any constructors, properties, attributes or methods for the class.

42. (Previously Presented) The method of claim 41, wherein if there are any constructors, properties, attributes or methods for the class, storing the constructors, properties, attributes or methods for the class in a temporary structure in preparation for storing in the first dictionary data structure.

43. (Previously Presented) The method of claim 36, wherein providing an indication of differences between the first and second binary software builds comprises providing a delta which indicates entries that have been added or deleted between the first dictionary data structure and the second dictionary data structure.

44. (Previously Presented) The method of claim 36, further comprising displaying a tree representation of the dictionary data structure that can be used to determine if any classes have been modified.

45. (Previously Presented) The method of claim 36, further comprising receiving user input for loading constructors to specific classes for increasing code coverage testing for changes between the first software build and the second software build.

46. (Previously Presented) A storage computer readable medium comprising computer executable instructions for performing the acts of claim 36.